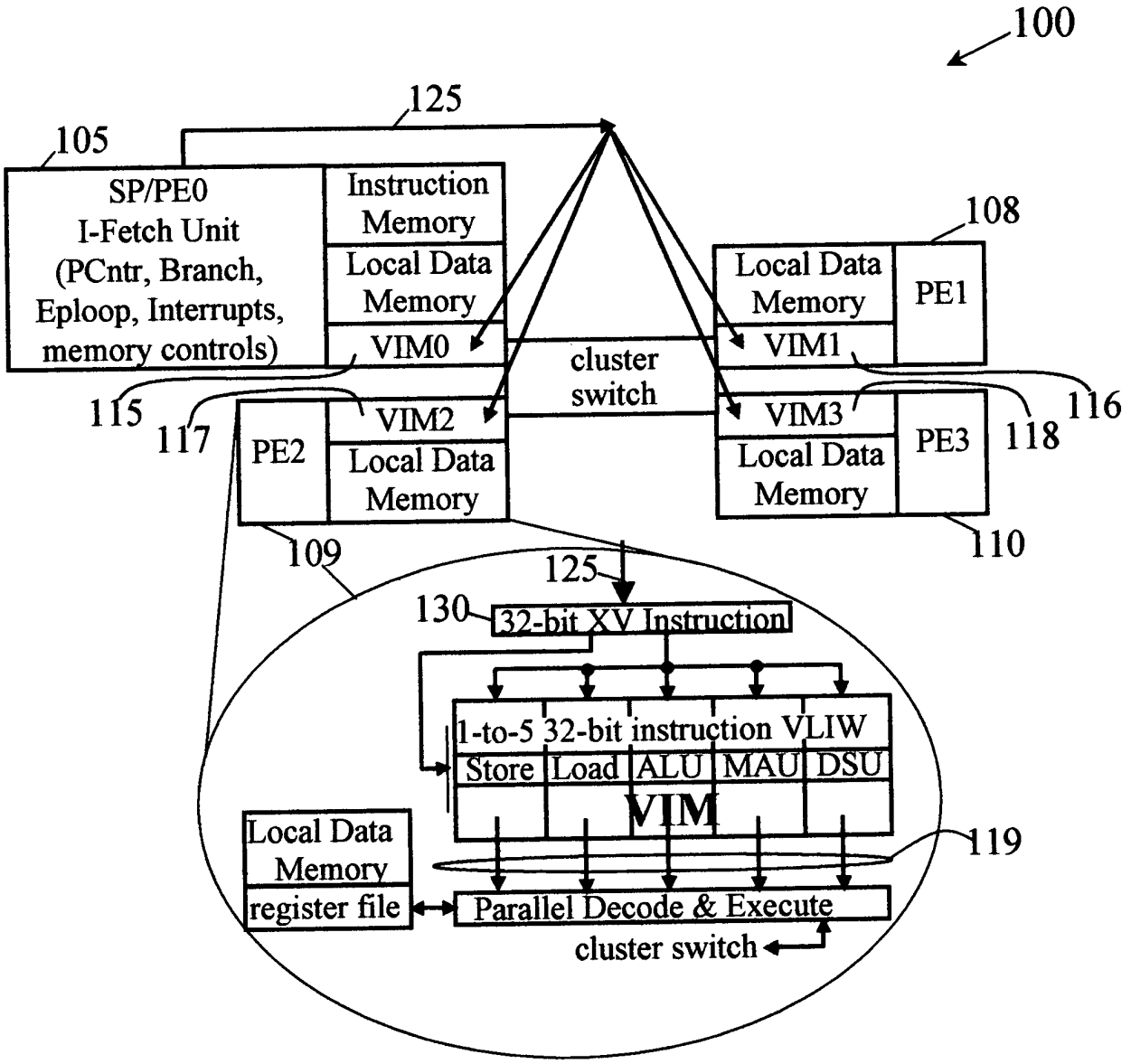


Fig. 1 (Prior Art)

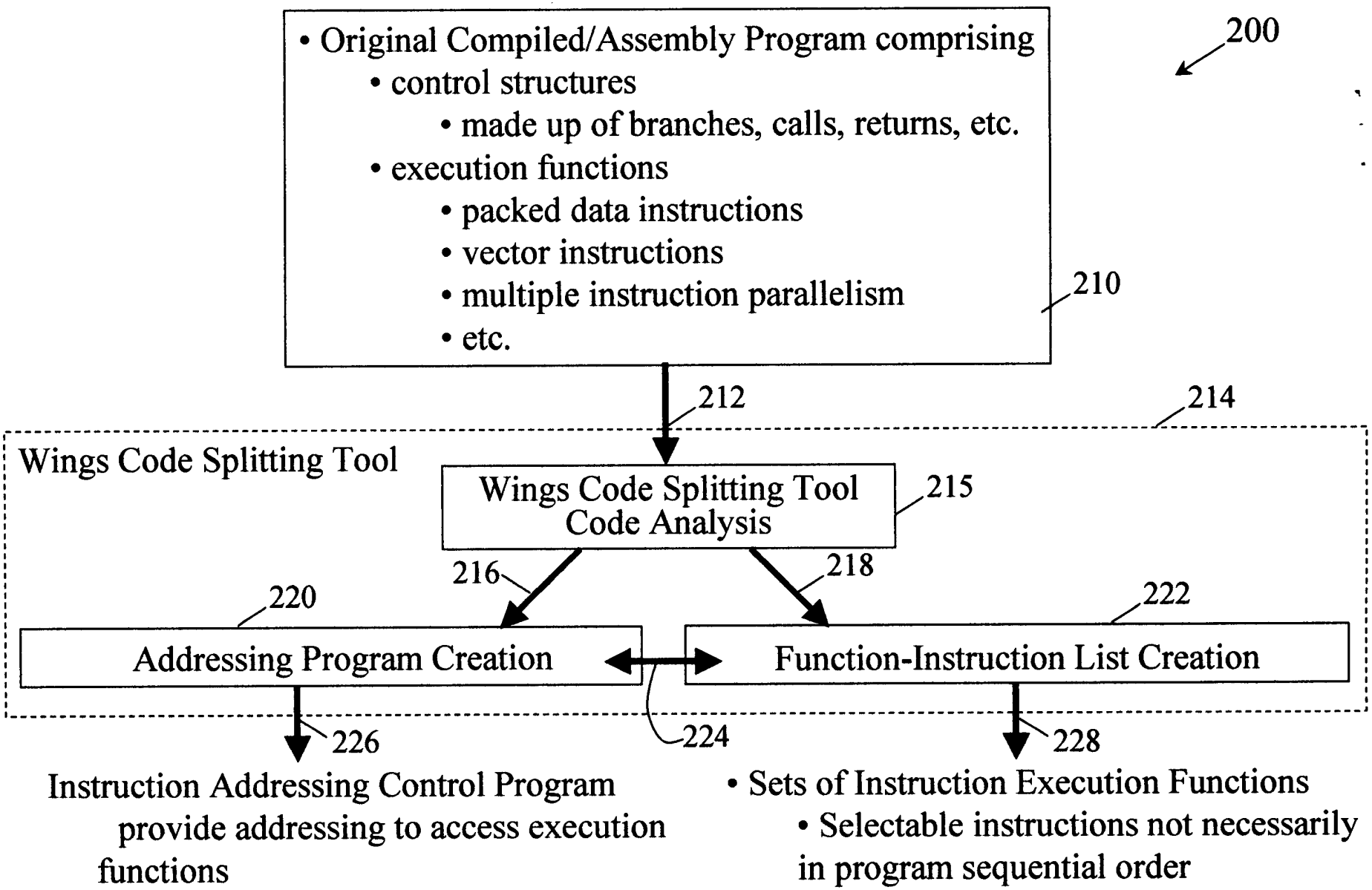

• Original Compiled/Assembly Program comprising
• control structures
• made up of branches, calls, returns, etc.
• execution functions
• packed data instructions
• vector instructions
• multiple instruction parallelism
• etc.
200
210
212
214
Wings Code Splitting Tool
Wings Code Splitting Tool Code Analysis
215
216
218
220
222
Addressing Program Creation
Function-Instruction List Creation
224
226
228
Instruction Addressing Control Program
provide addressing to access execution functions
• Sets of Instruction Execution Functions
• Selectable instructions not necessarily in program sequential order

Fig. 2

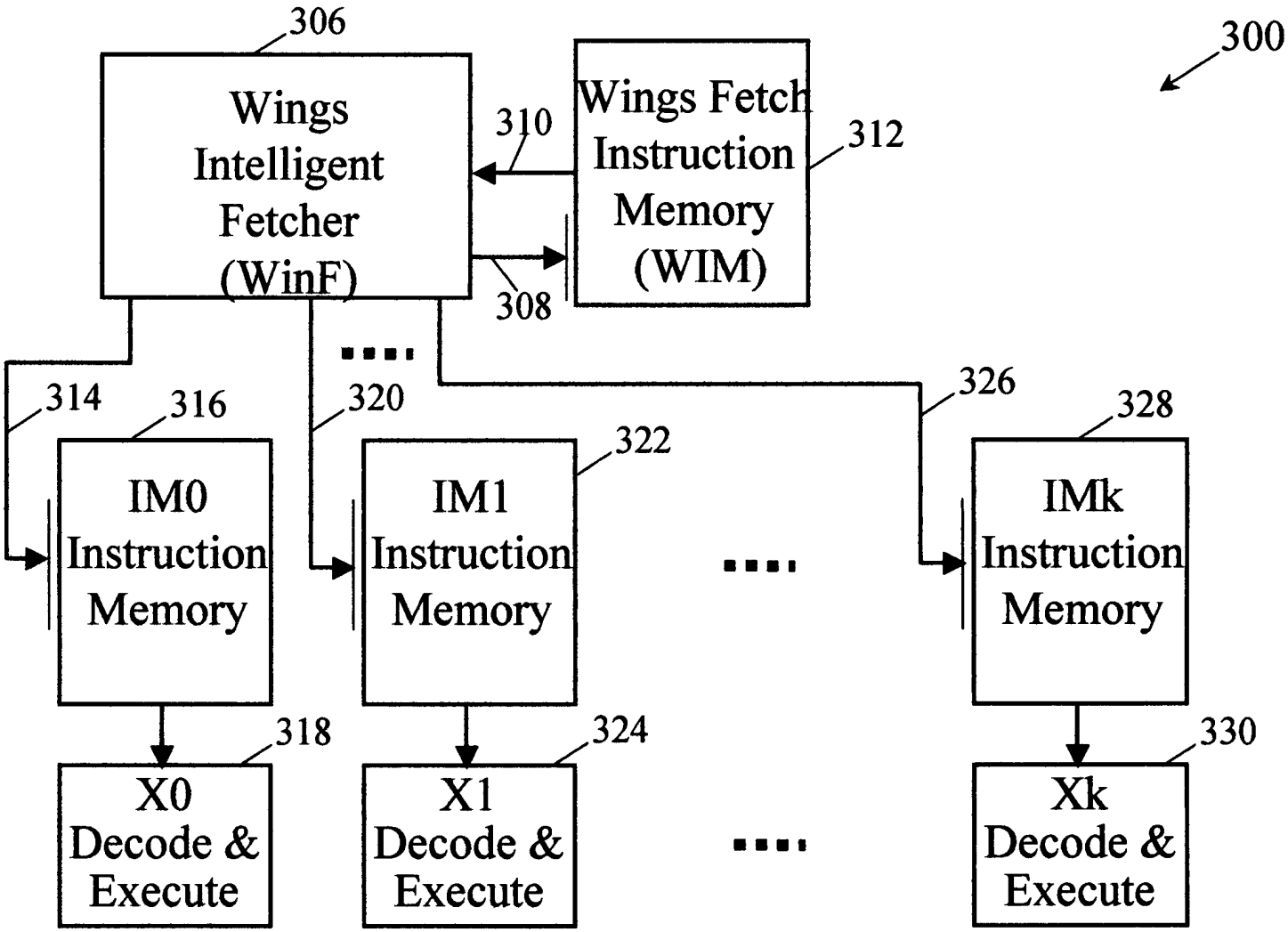

306
300
Wings
Intelligent
Fetcher
(WinF)
310
Wings Fetch
Instruction
Memory
(WIM)
312
308
314
316
320
326
328
IM0
Instruction
Memory
IM1
Instruction
Memory
322
IMk
Instruction
Memory
318
324
330
X0
Decode &
Execute
X1
Decode &
Execute
Xk
Decode &
Execute

Fig. 3

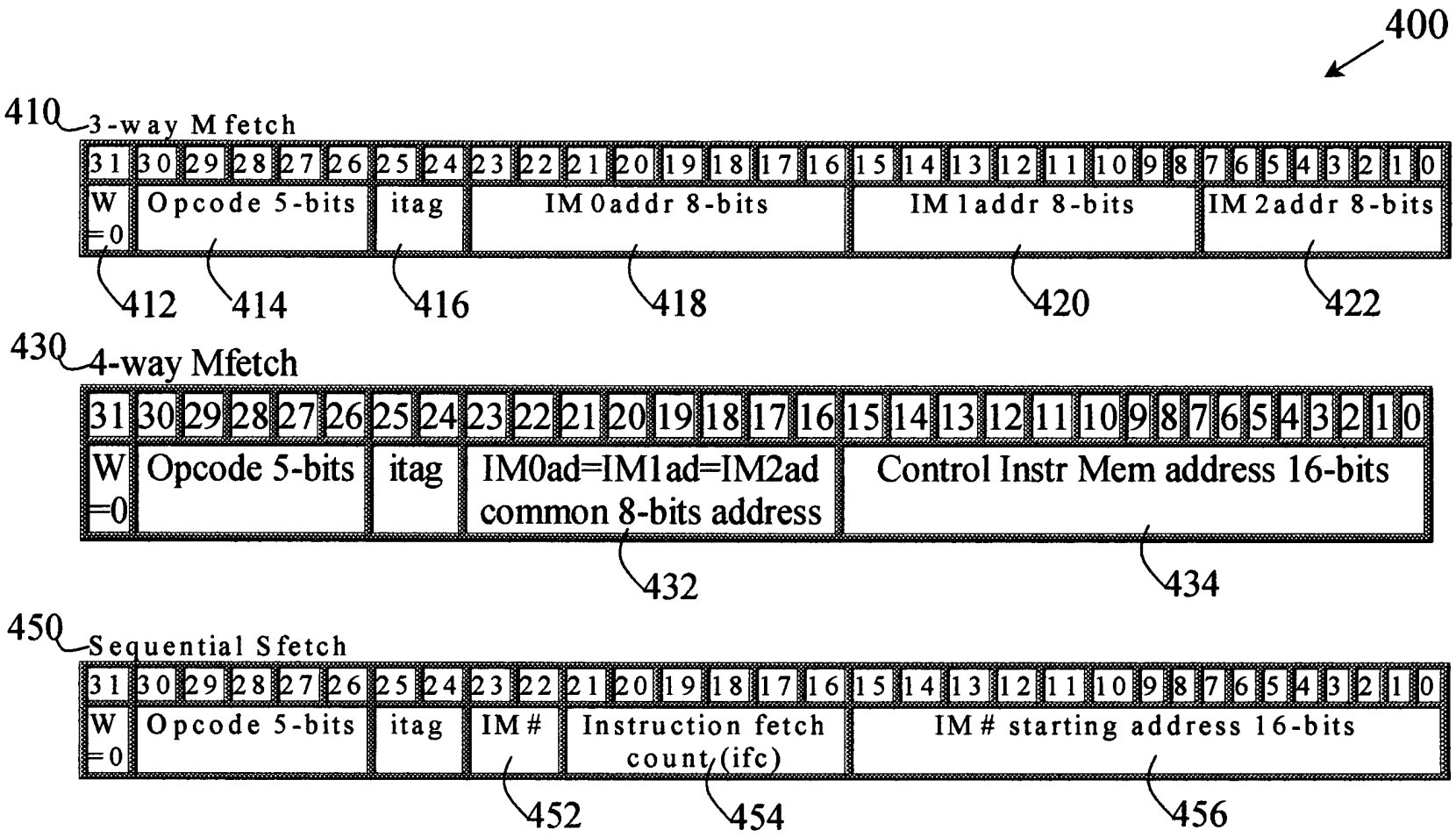

400
410
3-way M fetch
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
itag
IM 0addr 8-bits
IM 1addr 8-bits
IM 2addr 8-bits
412
414
416
418
420
422
430
4-way Mfetch
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
itag
IM0ad=IM1ad=IM2ad common 8-bits address
Control Instr Mem address 16-bits
432
434
450
Sequential Sfetch
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
itag
IM #
Instruction fetch count (ifc)
IM # starting address 16-bits
452
454
456

Fig. 4

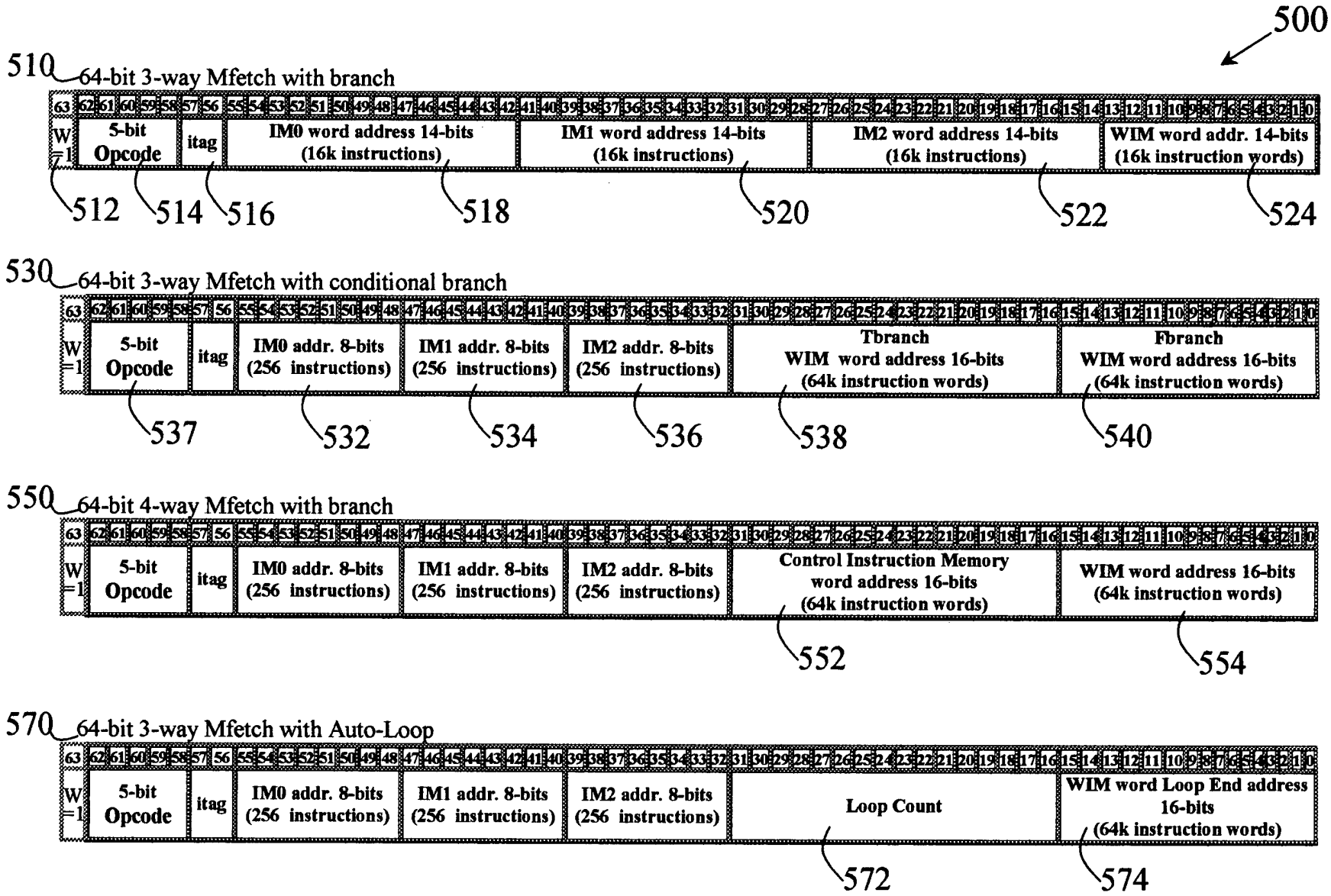

500
510
64-bit 3-way Mfetch with branch
W=1
5-bit Opcode
itag
IM0 word address 14-bits (16k instructions)
IM1 word address 14-bits (16k instructions)
IM2 word address 14-bits (16k instructions)
WIM word addr. 14-bits (16k instruction words)
512
514
516
518
520
522
524
530
64-bit 3-way Mfetch with conditional branch
W=1
5-bit Opcode
itag
IM0 addr. 8-bits (256 instructions)
IM1 addr. 8-bits (256 instructions)
IM2 addr. 8-bits (256 instructions)
Tbranch WIM word address 16-bits (64k instruction words)
Fbranch WIM word address 16-bits (64k instruction words)
537
532
534
536
538
540
550
64-bit 4-way Mfetch with branch
W=1
5-bit Opcode
itag
IM0 addr. 8-bits (256 instructions)
IM1 addr. 8-bits (256 instructions)
IM2 addr. 8-bits (256 instructions)
Control Instruction Memory word address 16-bits (64k instruction words)
WIM word address 16-bits (64k instruction words)
552
554
570
64-bit 3-way Mfetch with Auto-Loop
W=1
5-bit Opcode
itag
IM0 addr. 8-bits (256 instructions)
IM1 addr. 8-bits (256 instructions)
IM2 addr. 8-bits (256 instructions)
Loop Count
WIM word Loop End address 16-bits (64k instruction words)
572
574

Fig. 5

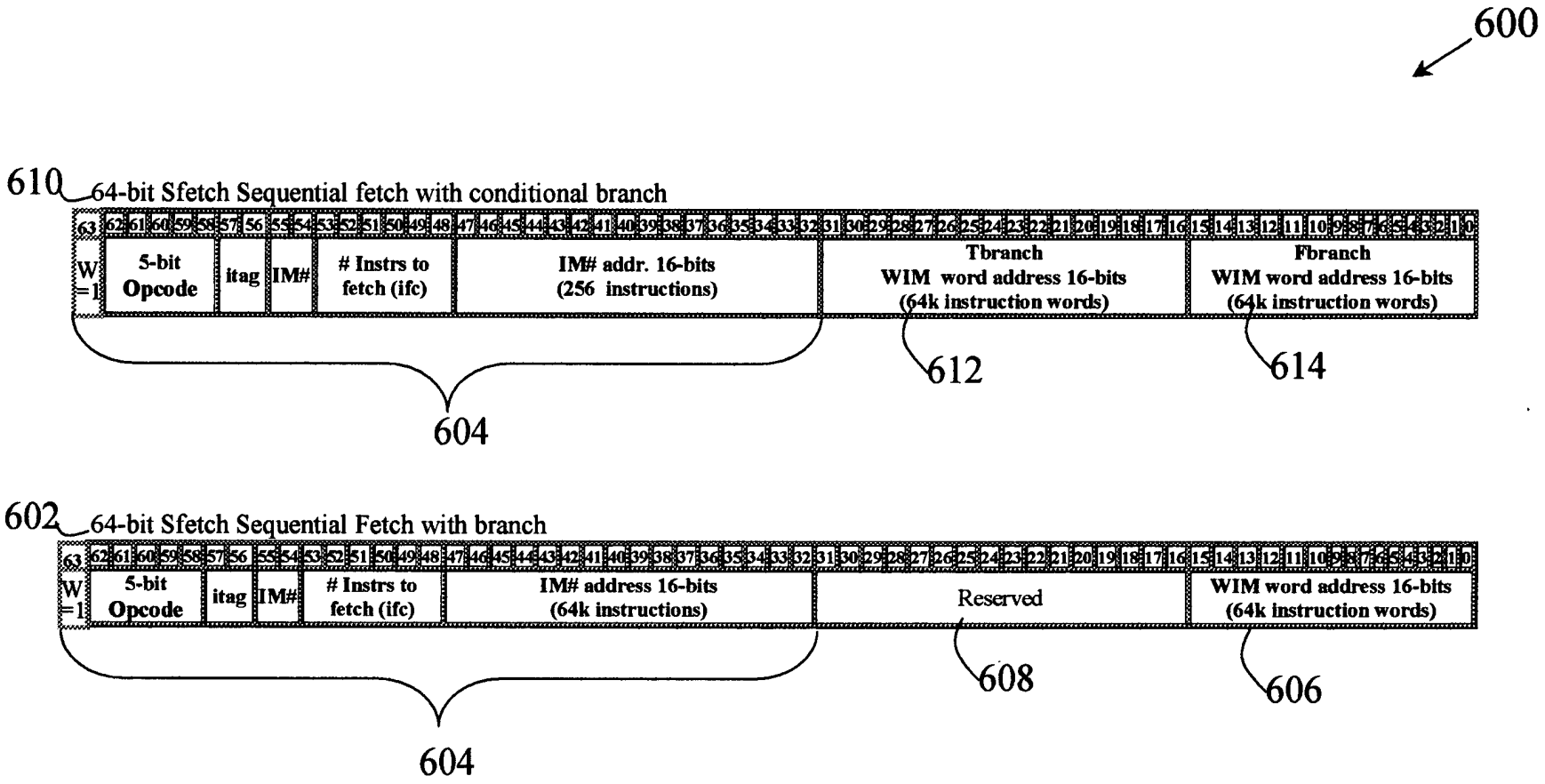

600
610
64-bit Sfetch Sequential fetch with conditional branch
63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W=1
5-bit Opcode
itag
IM#
Instrs to fetch (ifc)
IM# addr. 16-bits (256 instructions)
Tbranch WIM word address 16-bits (64k instruction words)
Fbranch WIM word address 16-bits (64k instruction words)
612
614
604
602
64-bit Sfetch Sequential Fetch with branch
63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W=1
5-bit Opcode
itag
IM#
Instrs to fetch (ifc)
IM# address 16-bits (64k instructions)
Reserved
WIM word address 16-bits (64k instruction words)
608
606
604

Fig. 6A

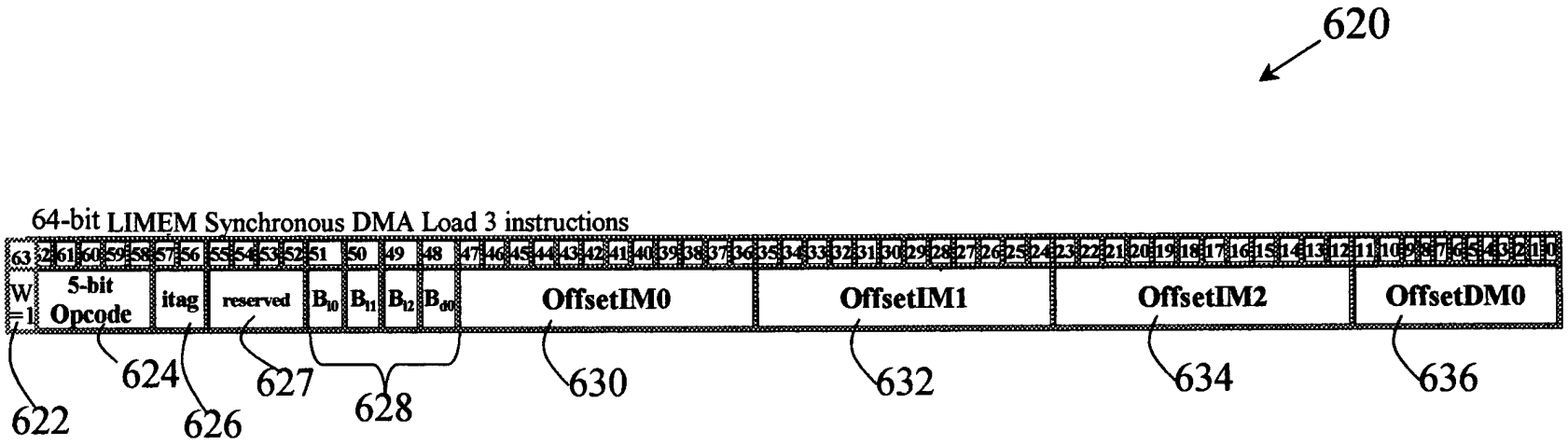

620
64-bit LIMEM Synchronous DMA Load 3 instructions
63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W=1
5-bit Opcode
itag
reserved
B10
B11
B12
B60
OffsetIM0
OffsetIM1
OffsetIM2
OffsetDM0
622
624
626
627
628
630
632
634
636

Fig. 6B

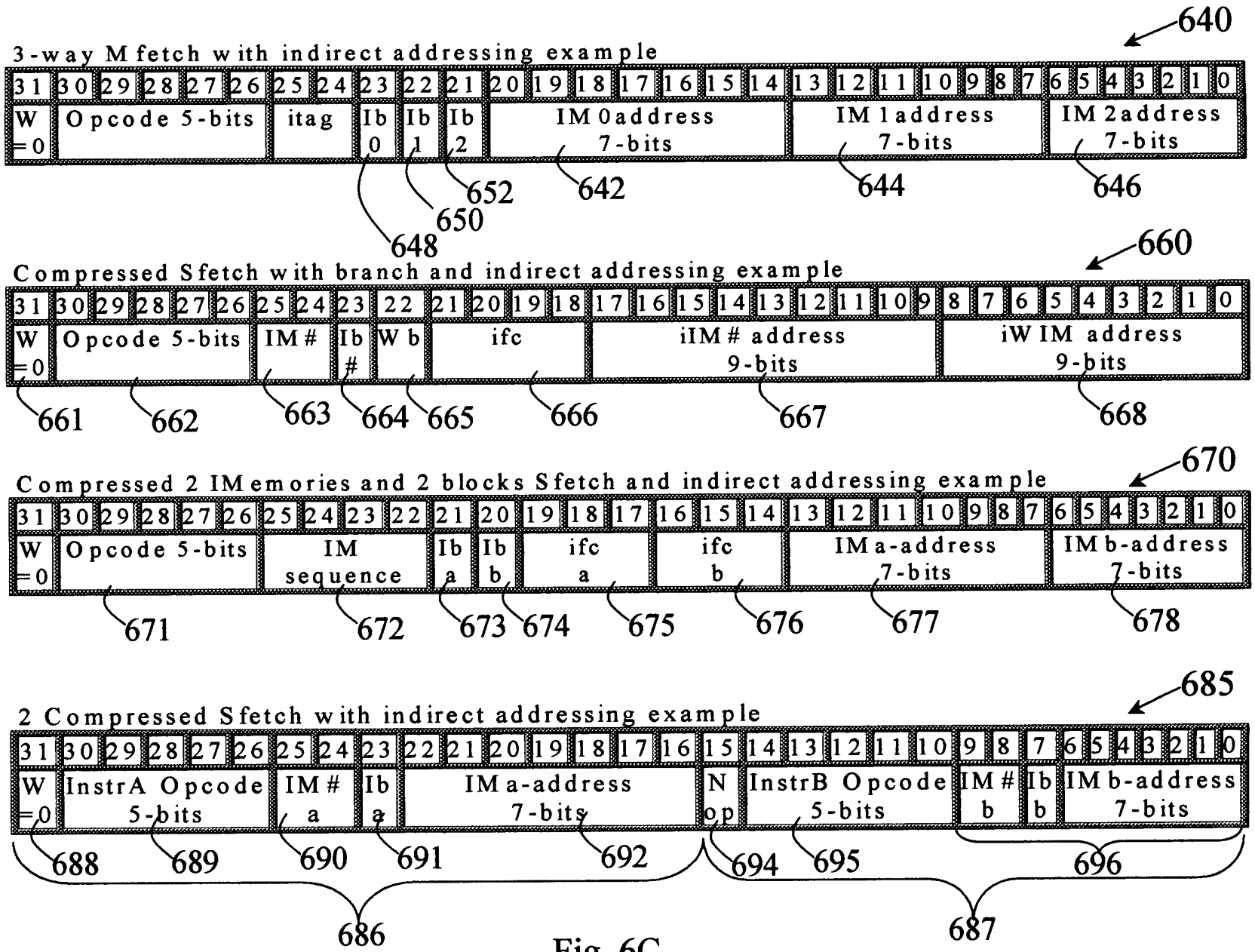

640
3-way M fetch with indirect addressing example
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
itag
Ib 0
Ib 1
Ib 2
IM 0address 7-bits
IM 1address 7-bits
IM 2address 7-bits
652
642
644
646
650
648
660
Compressed Sfetch with branch and indirect addressing example
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
IM #
Ib #
W b
ifc
iIM # address 9-bits
iW IM address 9-bits
661
662
663
664
665
666
667
668
670
Compressed 2 IMemories and 2 blocks Sfetch and indirect addressing example
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
Opcode 5-bits
IM sequence
Ib a
Ib b
ifc a
ifc b
IM a-address 7-bits
IM b-address 7-bits
671
672
673
674
675
676
677
678
685
2 Compressed Sfetch with indirect addressing example
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
W =0
InstrA Opcode 5-bits
IM # a
Ib a
IM a-address 7-bits
N op
InstrB Opcode 5-bits
IM # b
Ib b
IM b-address 7-bits
688
689
690
691
692
694
695
696
686
687

Fig. 6C

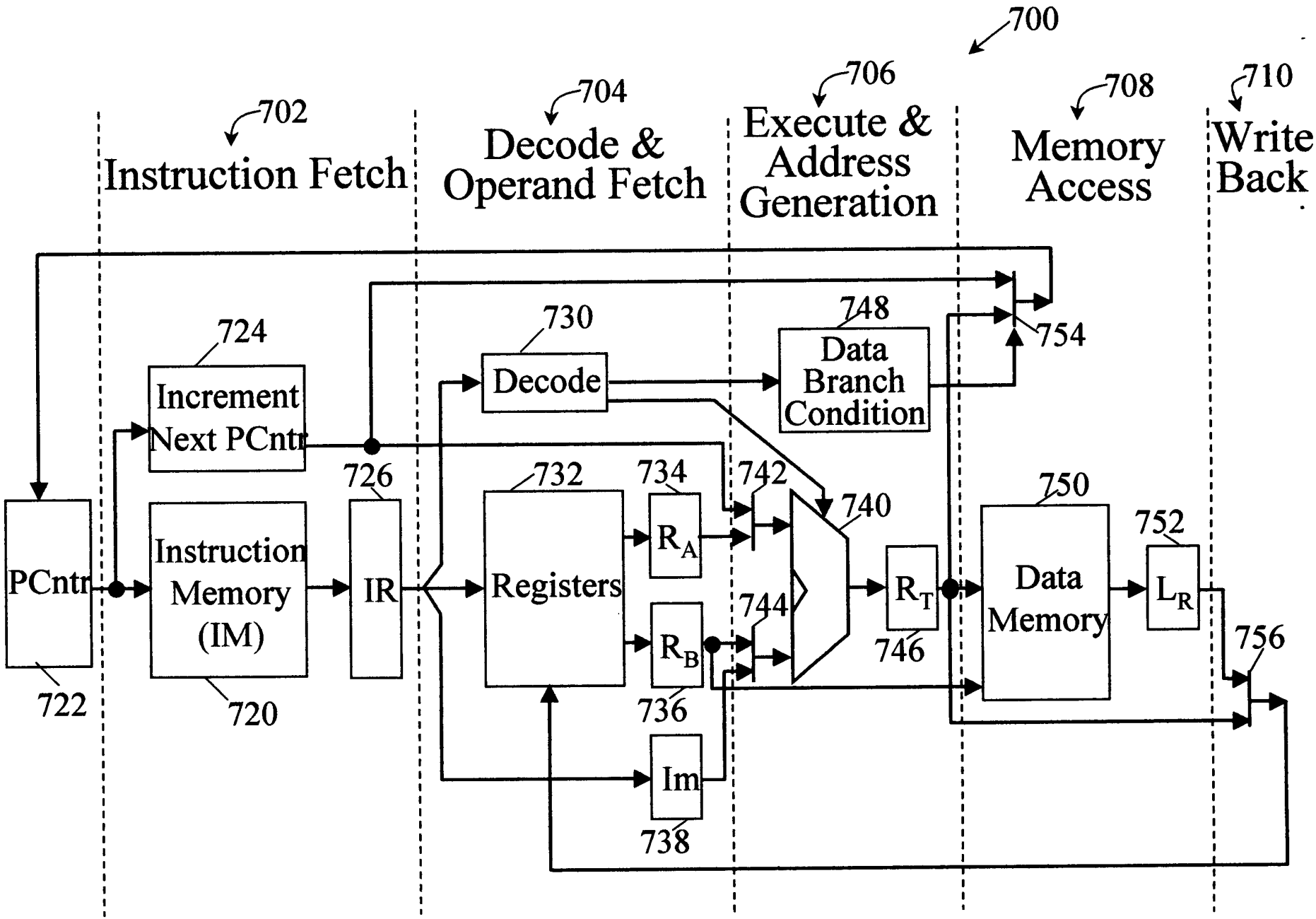


Fig. 7 (Prior Art)

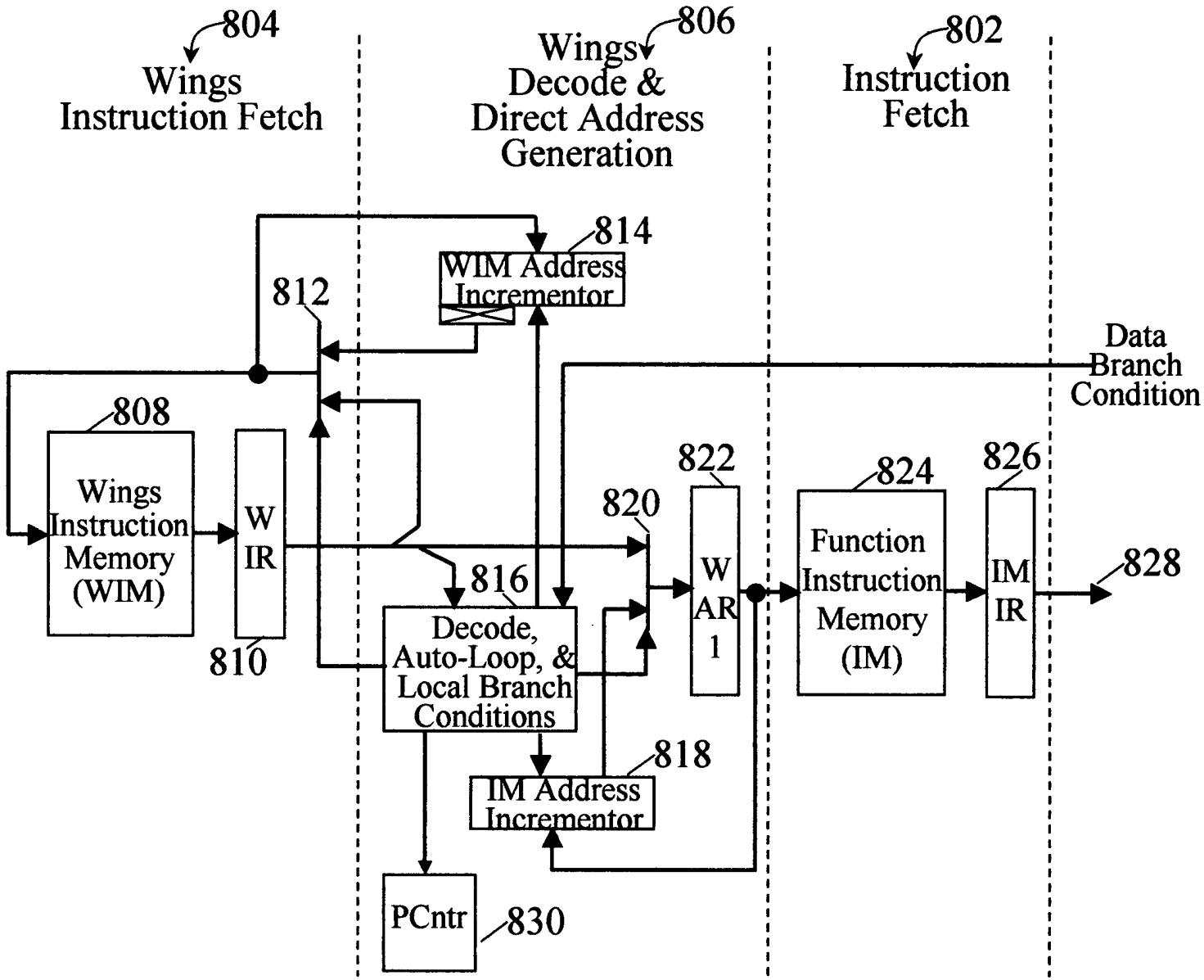

804
Wings
Instruction Fetch
806
Wings
Decode &
Direct Address
Generation
802
Instruction
Fetch
800
814
WIM Address
Incrementor
812
Data
Branch
Condition
808
Wings
Instruction
Memory
(WIM)
W
IR
810
816
Decode,
Auto-Loop, &
Local Branch
Conditions
820
822
W
AR
1
824
Function
Instruction
Memory
(IM)
826
IM
IR
828
818
IM Address
Incrementor
PCntr
830

Fig. 8

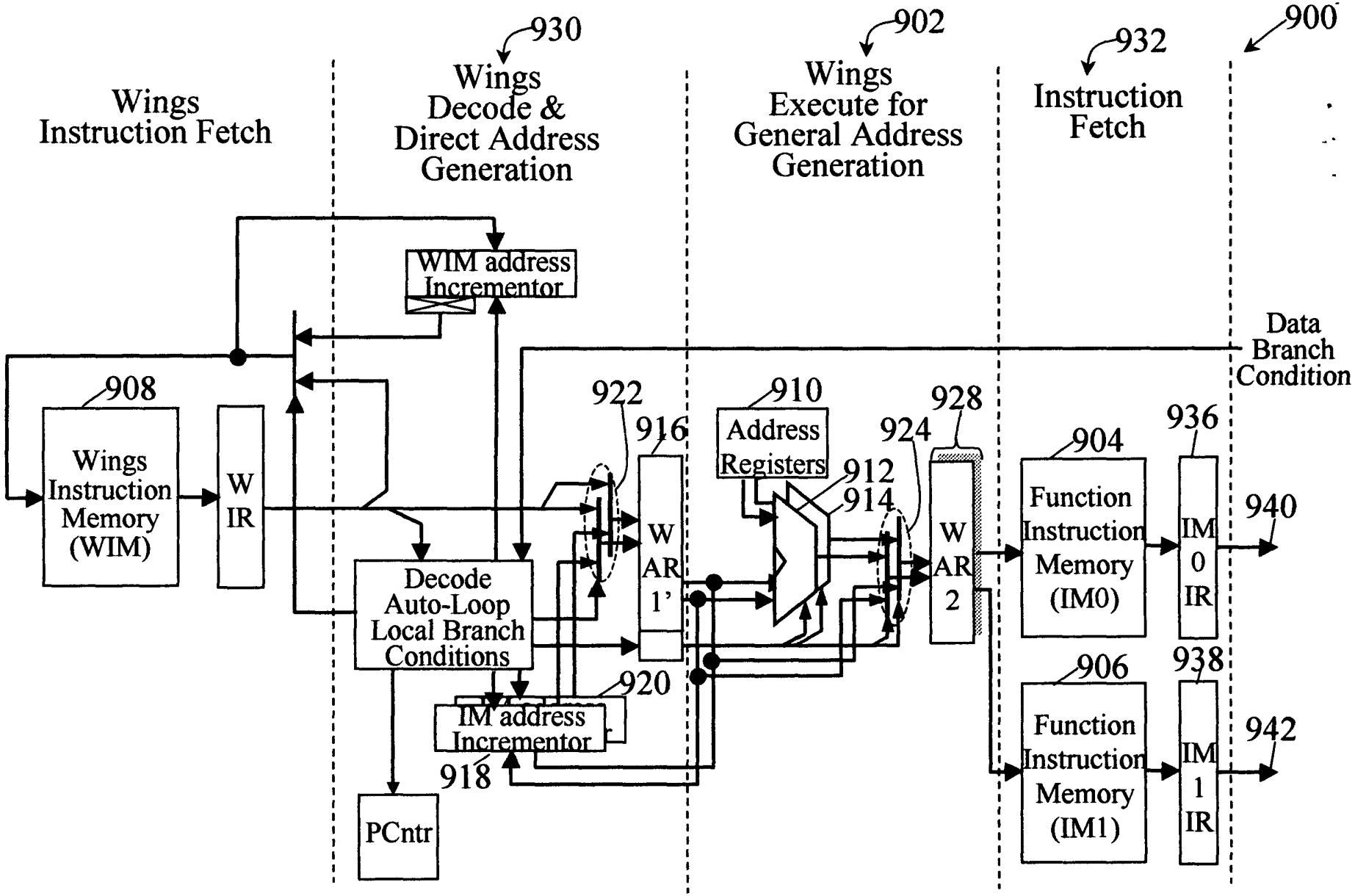

930
Wings
Instruction Fetch
Wings
Decode &
Direct Address
Generation
902
Wings
Execute for
General Address
Generation
932
Instruction
Fetch
900
WIM address
Incrementor
Data
Branch
Condition
908
Wings
Instruction
Memory
(WIM)
W
IR
922
916
910
Address
Registers
924
928
912
914
936
904
Function
Instruction
Memory
(IM0)
IM
0
IR
940
W
AR
1'
W
AR
2
Decode
Auto-Loop
Local Branch
Conditions
920
IM address
Incrementor
918
906
938
Function
Instruction
Memory
(IM1)
IM
1
IR
942
PCntr

Fig. 9

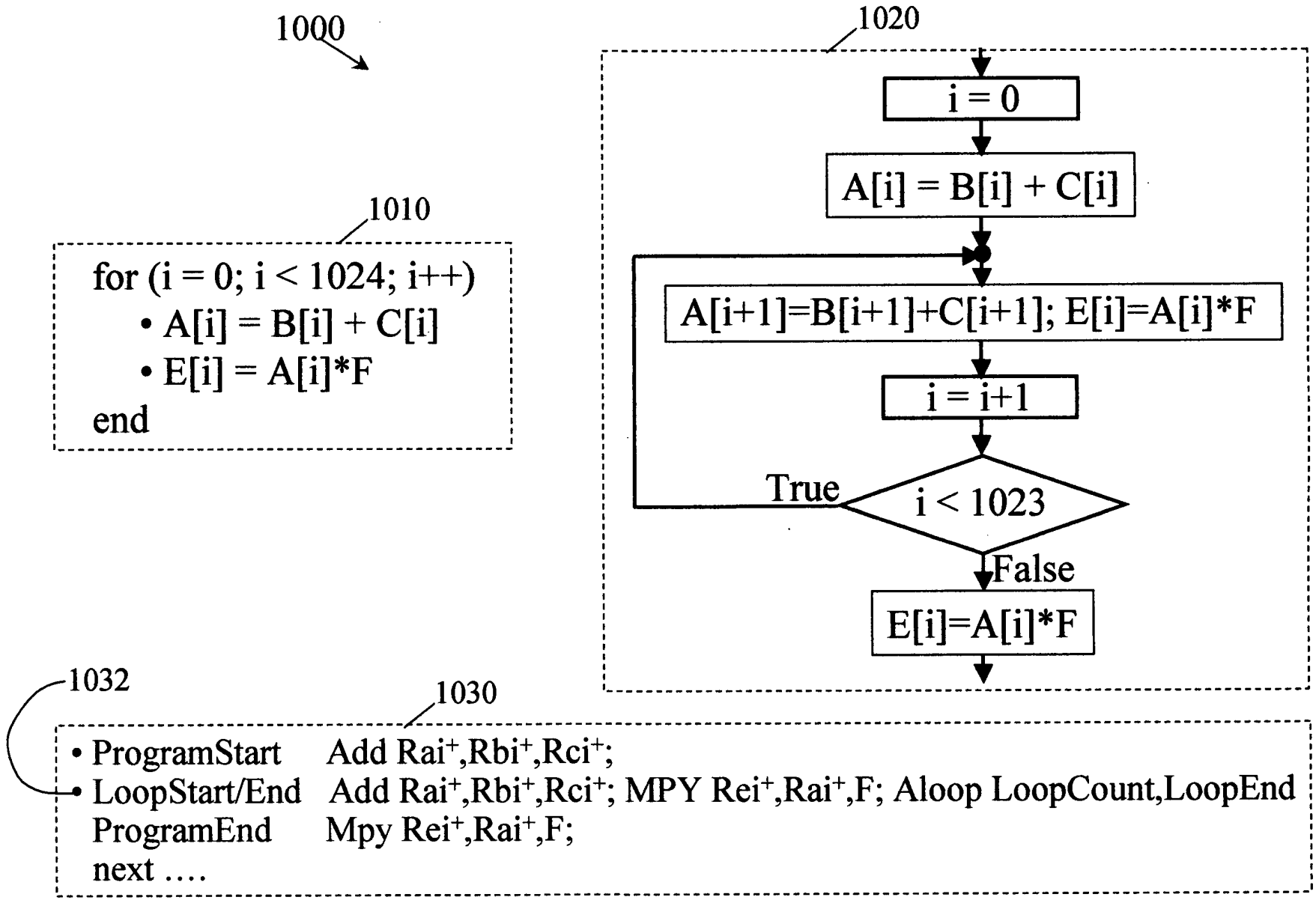
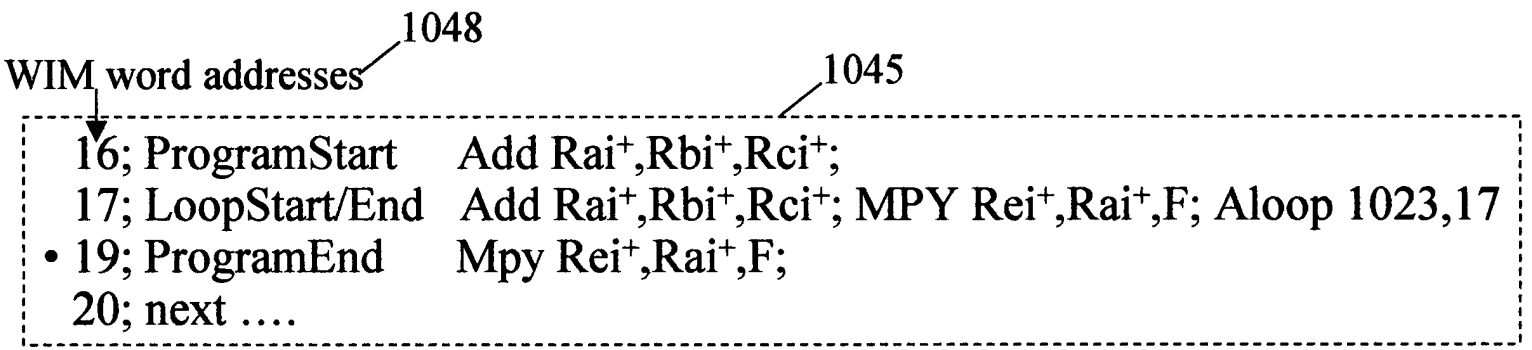

1000
1010
for (i = 0; i < 1024; i++)
• A[i] = B[i] + C[i]
• E[i] = A[i]*F
end
1020
i = 0
A[i] = B[i] + C[i]
A[i+1]=B[i+1]+C[i+1]; E[i]=A[i]*F
i = i+1
True
i < 1023
False
E[i]=A[i]*F
1032
1030
• ProgramStart Add Rai+,Rbi+,Rci+;
• LoopStart/End Add Rai+,Rbi+,Rci+; MPY Rei+,Rai+,F; Aloop LoopCount,LoopEnd
ProgramEnd Mpy Rei+,Rai+,F;
next ....

Fig. 10A



1050

	1052 Clock	1054 Fetch WIM Address	1056 Decode WIM instruction	1058 Execute Gen Fetch Address	1060 IM0 Addr	1062 IM0 Instruction	1064 IM1 Addr	1066 IM1 Instruction
1068	j	16						
1070	j+1	17, 18	0Sfetch 26, n, n*;					
1072	j+2	n	1Mfetch 26, 11, n; Aloop 1023, 17;	26, n, n				
1074	j+3	n	1Mfetch 26, 11, n;	26, 11, n	26	Add Rai+,Rbi+,Rci+		
1076	j+4	n	1Mfetch 26, 11, n;	26, 11, n	26	Add Rai+,Rbi+,Rci+	11	Mpy Rei+,Rai+,F
	j+5	n	1Mfetch 26, 11, n;	26, 11, n	26	Add Rai+,Rbi+,Rci+	11	Mpy Rei+,Rai+,F
	j+6	n	1Mfetch 26, 11, n;	26, 11, n	26	Add Rai+,Rbi+,Rci+	11	Mpy Rei+,Rai+,F
	:	:	:	:	:	:	:	:
1082	j+1024	19	0Sfetch n, 11, n;	26, 11, n	26	Add Rai+,Rbi+,Rci+	11	Mpy Rei+,Rai+,F
1084	j+1025	20	0Nexti a1,a2,a3	n, 11, n	26	Add Rai+,Rbi+,Rci+	11	Mpy Rei+,Rai+,F
1086	j+1026	21	:	a1,a2,a3			11	Mpy Rei+,Rai+,F
1088	j+1027	22	:	:	a1	IM0nexti	a2	IM1nexti

*n=not used

Fig. 10B

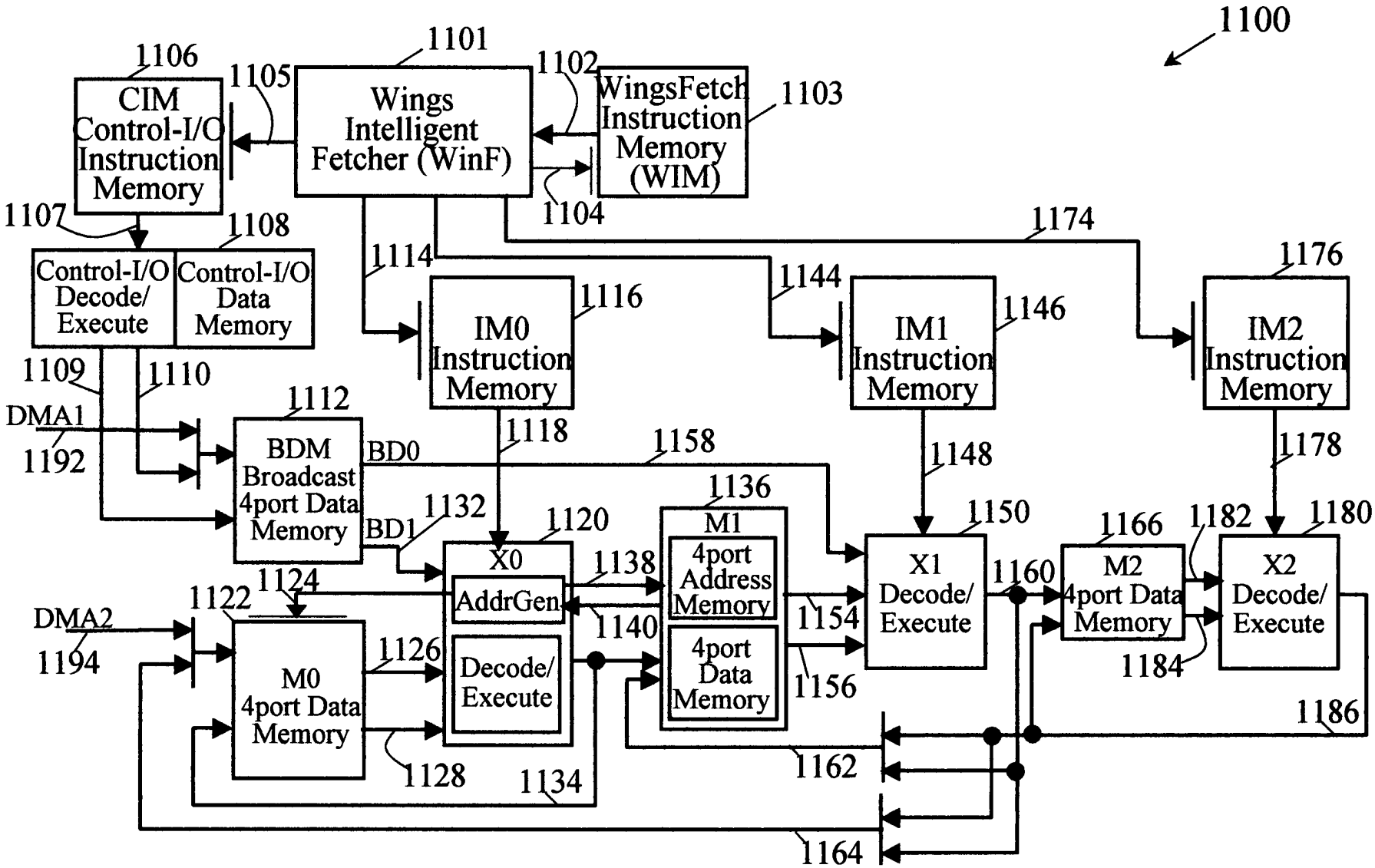
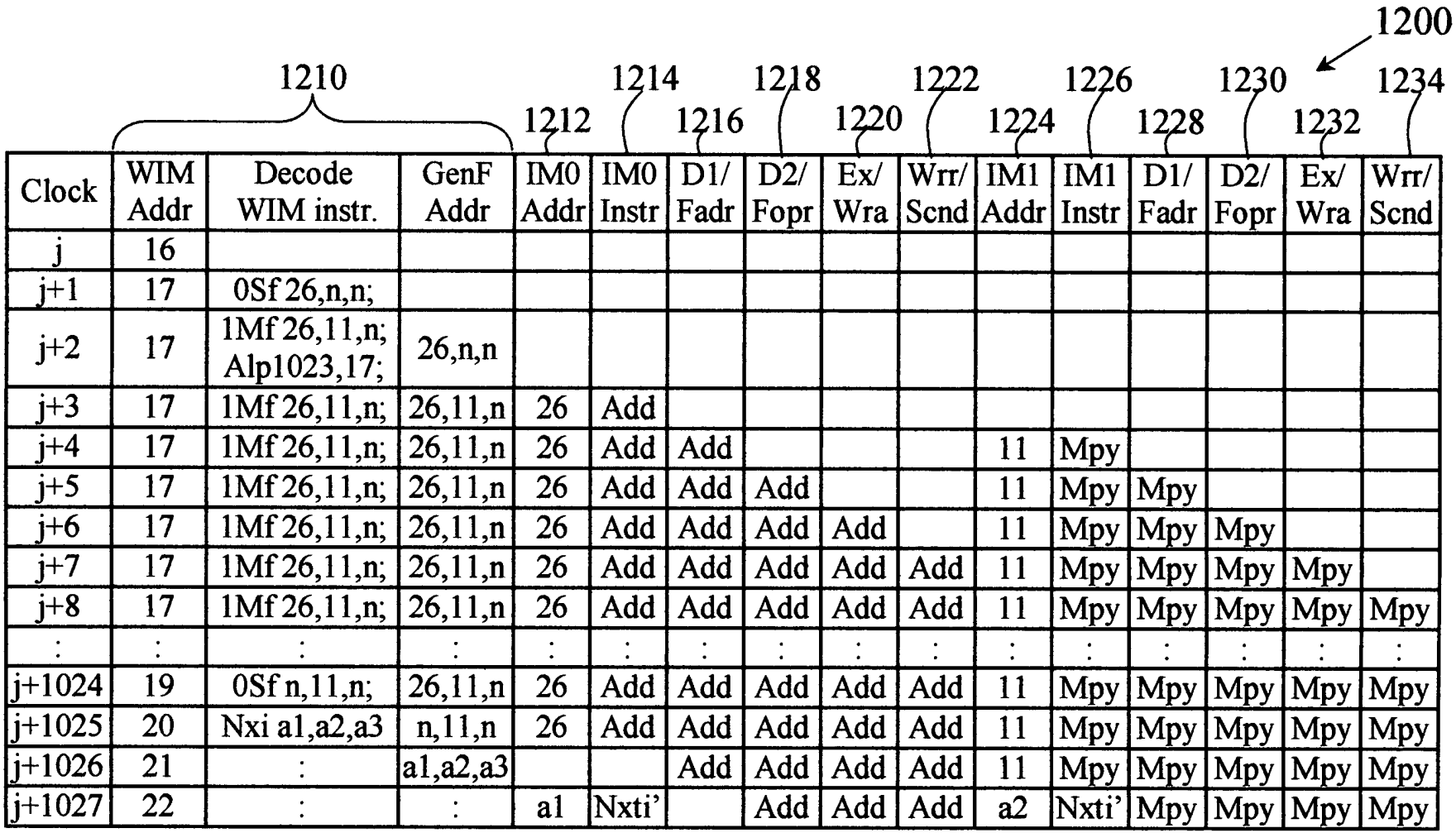

1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1112
1114
1116
1118
1120
1122
1124
1126
1128
1132
1134
1136
1138
1140
1144
1146
1148
1150
1154
1156
1158
1160
1162
1164
1166
1174
1176
1178
1180
1182
1184
1186
1192
1194
CIM Control-I/O Instruction Memory
Wings Intelligent Fetcher (WinF)
WingsFetch Instruction Memory (WIM)
Control-I/O Decode/ Execute
Control-I/O Data Memory
IM0 Instruction Memory
IM1 Instruction Memory
IM2 Instruction Memory
DMA1
DMA2
BDM Broadcast 4port Data Memory
BD0
BD1
X0
AddrGen
Decode/ Execute
M0 4port Data Memory
M1
4port Address Memory
4port Data Memory
X1 Decode/ Execute
M2 4port Data Memory
X2 Decode/ Execute

Fig. 11



Clock	WIM Addr	Decode WIM instr.	GenF Addr	IM0 Addr	IM0 Instr	D1/ Fadr	D2/ Fopr	Ex/ Wra	Wrr/ Scnd	IM1 Addr	IM1 Instr	D1/ Fadr	D2/ Fopr	Ex/ Wra	Wrr/ Scnd
j	16														
j+1	17	0Sf 26,n,n;													
j+2	17	1Mf 26,11,n; Alp1023,17;	26,n,n												
j+3	17	1Mf 26,11,n;	26,11,n	26	Add										
j+4	17	1Mf 26,11,n;	26,11,n	26	Add	Add				11	Mpy				
j+5	17	1Mf 26,11,n;	26,11,n	26	Add	Add	Add			11	Mpy	Mpy			
j+6	17	1Mf 26,11,n;	26,11,n	26	Add	Add	Add	Add		11	Mpy	Mpy	Mpy		
j+7	17	1Mf 26,11,n;	26,11,n	26	Add	Add	Add	Add	Add	11	Mpy	Mpy	Mpy	Mpy	
j+8	17	1Mf 26,11,n;	26,11,n	26	Add	Add	Add	Add	Add	11	Mpy	Mpy	Mpy	Mpy	Mpy
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
j+1024	19	0Sf n,11,n;	26,11,n	26	Add	Add	Add	Add	Add	11	Mpy	Mpy	Mpy	Mpy	Mpy
j+1025	20	Nxi a1,a2,a3	n,11,n	26	Add	Add	Add	Add	Add	11	Mpy	Mpy	Mpy	Mpy	Mpy
j+1026	21	:	a1,a2,a3			Add	Add	Add	Add	11	Mpy	Mpy	Mpy	Mpy	Mpy
j+1027	22	:	:	a1	Nxti'		Add	Add	Add	a2	Nxti'	Mpy	Mpy	Mpy	Mpy

- IM#Addr & IM#Instr = Fetch IM# instruction at IM# address
- D1/Fadr = Decode 1 & Fetch address registers
- D2/Fopr = Decode 2, Fetch operands & generate next operand address
- Ex/Wra = Execute & Write back operand addresses
- Wrr/Scnd = Write back results & Set condition registers
- Assumes operand bypassing is used

Fig. 12

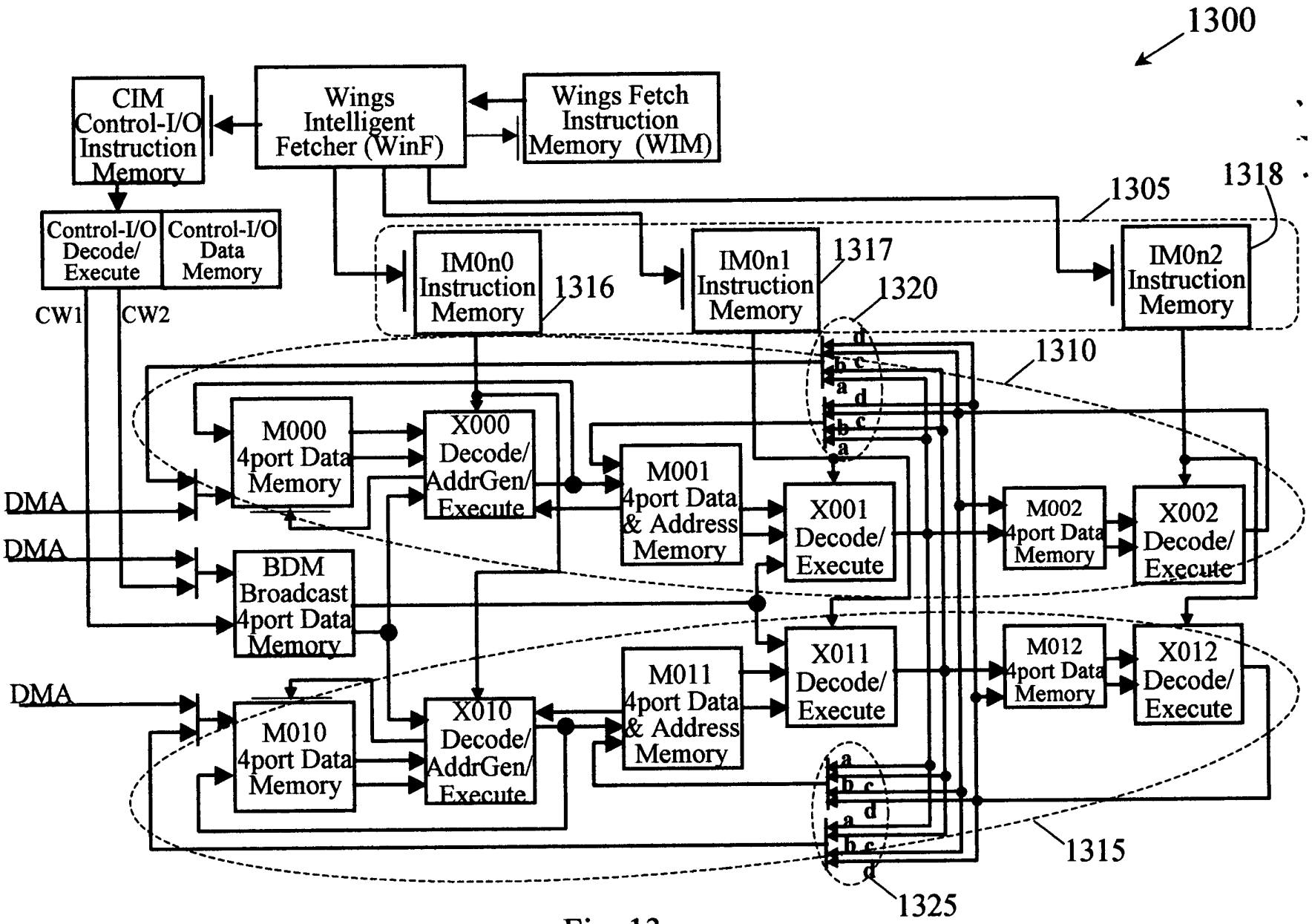

1300
CIM Control-I/O Instruction Memory
Wings Intelligent Fetcher (WinF)
Wings Fetch Instruction Memory (WIM)
1305
1318
Control-I/O Decode/ Execute
Control-I/O Data Memory
CW1
CW2
IM0n0 Instruction Memory
1316
IM0n1 Instruction Memory
1317
1320
IM0n2 Instruction Memory
1310
M000 4port Data Memory
X000 Decode/ AddrGen/ Execute
M001 4port Data & Address Memory
X001 Decode/ Execute
M002 4port Data Memory
X002 Decode/ Execute
DMA
DMA
BDM Broadcast 4port Data Memory
DMA
M010 4port Data Memory
X010 Decode/ AddrGen/ Execute
M011 4port Data & Address Memory
X011 Decode/ Execute
M012 4port Data Memory
X012 Decode/ Execute
1315
1325

Fig. 13

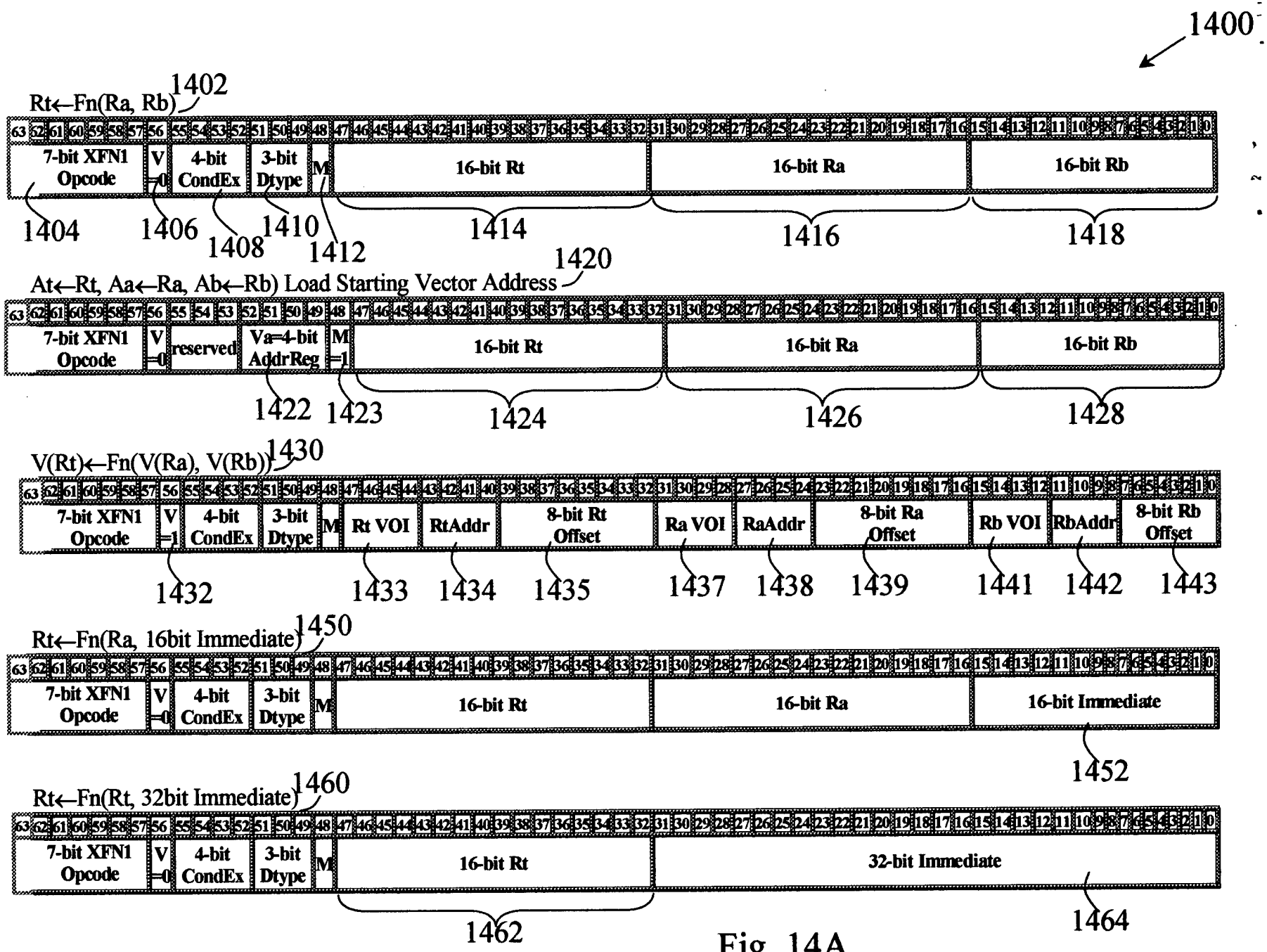

1400
Rt←Fn(Ra, Rb)
1402
7-bit XFN1 Opcode
V =0
4-bit CondEx
3-bit Dtype
M
16-bit Rt
16-bit Ra
16-bit Rb
1404
1406
1408
1410
1412
1414
1416
1418
At←Rt, Aa←Ra, Ab←Rb) Load Starting Vector Address
1420
7-bit XFN1 Opcode
V =0
reserved
Va=4-bit AddrReg
M =1
16-bit Rt
16-bit Ra
16-bit Rb
1422
1423
1424
1426
1428
V(Rt)←Fn(V(Ra), V(Rb))
1430
7-bit XFN1 Opcode
V =1
4-bit CondEx
3-bit Dtype
M
Rt VOI
RtAddr
8-bit Rt Offset
Ra VOI
RaAddr
8-bit Ra Offset
Rb VOI
RbAddr
8-bit Rb Offset
1432
1433
1434
1435
1437
1438
1439
1441
1442
1443
Rt←Fn(Ra, 16bit Immediate)
1450
7-bit XFN1 Opcode
V =0
4-bit CondEx
3-bit Dtype
M
16-bit Rt
16-bit Ra
16-bit Immediate
1452
Rt←Fn(Rt, 32bit Immediate)
1460
7-bit XFN1 Opcode
V =0
4-bit CondEx
3-bit Dtype
M
16-bit Rt
32-bit Immediate
1462
1464

Fig. 14A

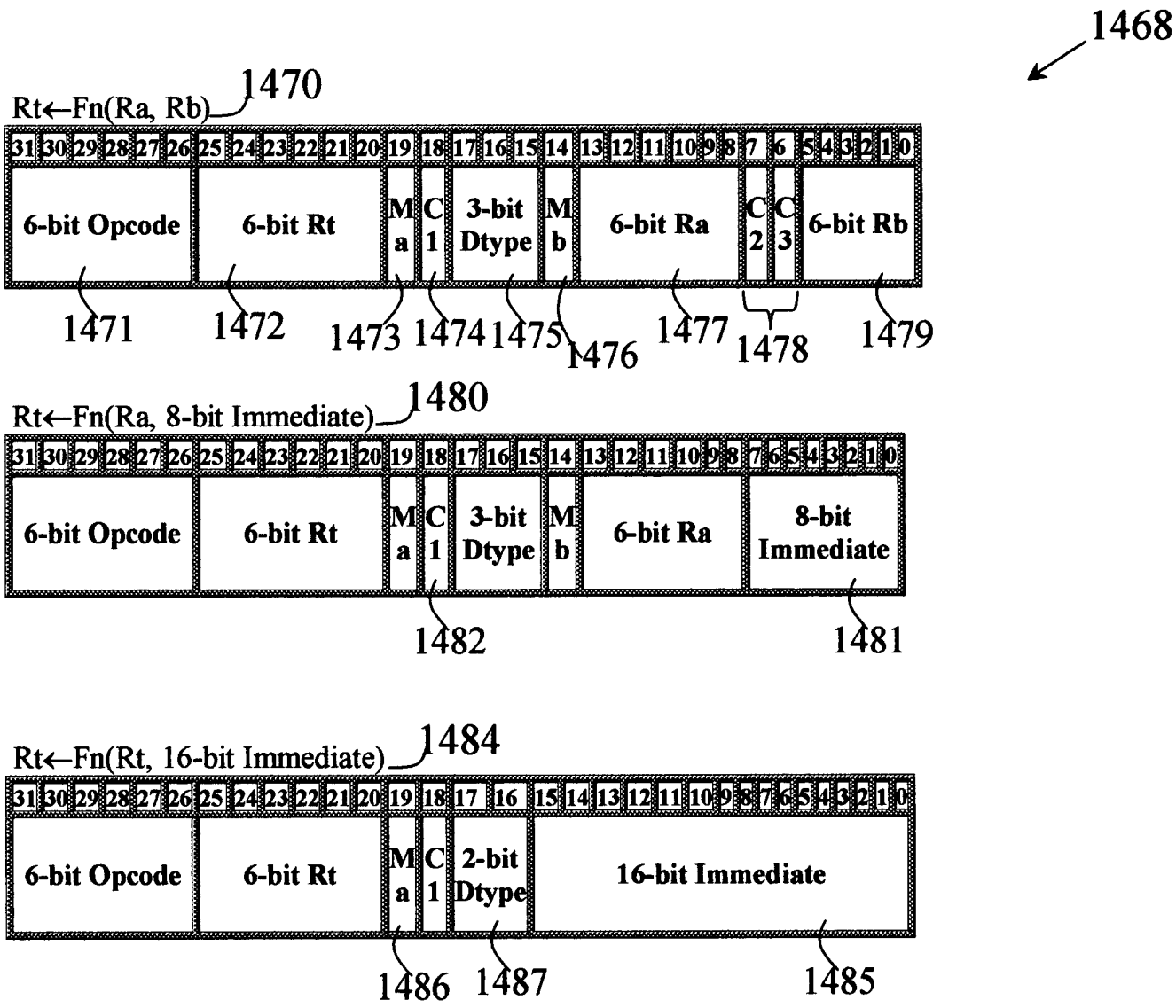

1468
Rt←Fn(Ra, Rb)
1470
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
6-bit Rt
Ma
C1
3-bit Dtype
Mb
6-bit Ra
C2
C3
6-bit Rb
1471
1472
1473
1474
1475
1476
1477
1478
1479
Rt←Fn(Ra, 8-bit Immediate)
1480
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
6-bit Rt
Ma
C1
3-bit Dtype
Mb
6-bit Ra
8-bit Immediate
1482
1481
Rt←Fn(Rt, 16-bit Immediate)
1484
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
6-bit Rt
Ma
C1
2-bit Dtype
16-bit Immediate
1486
1487
1485

Fig. 14B

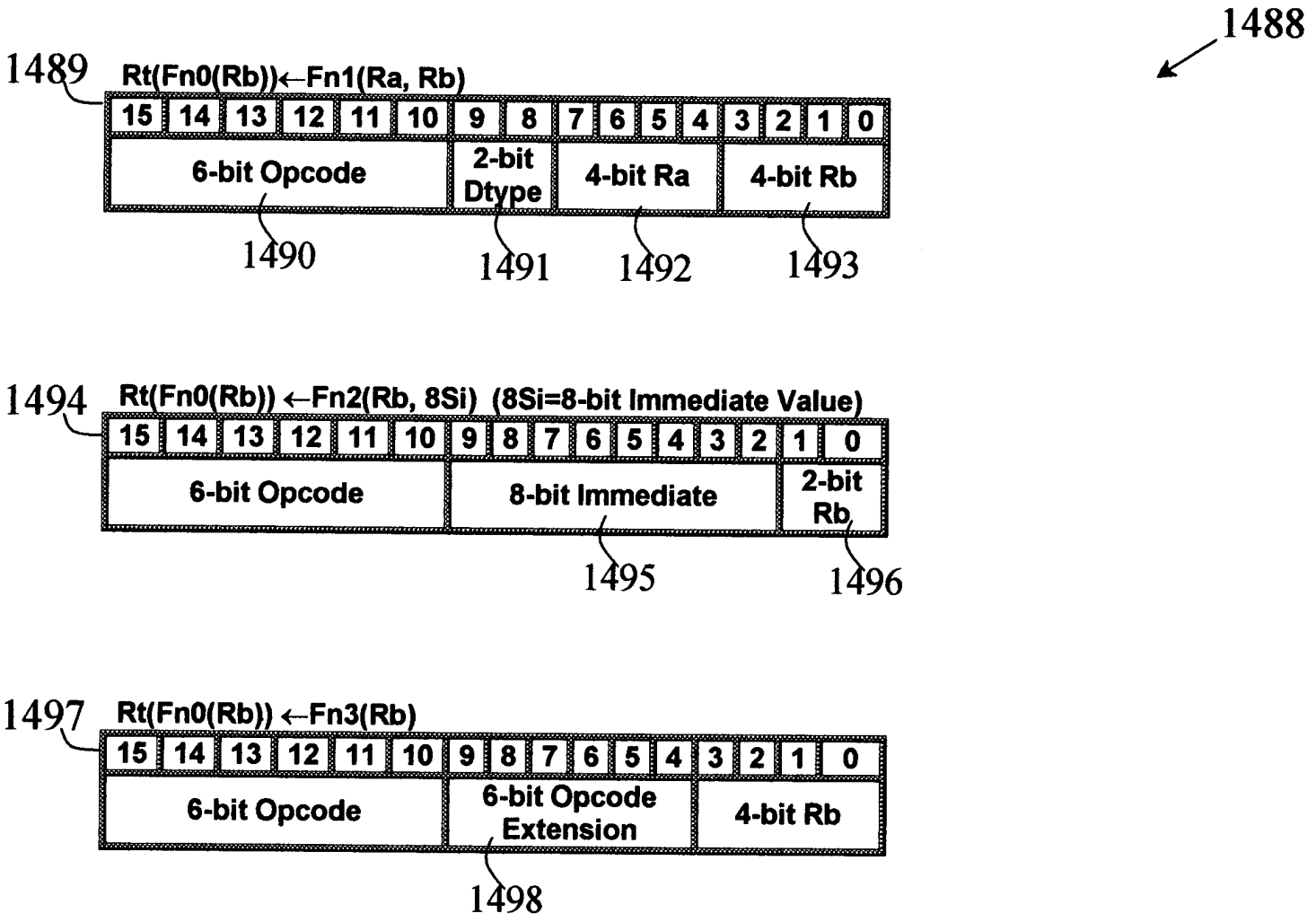

1488
1489
Rt(Fn0(Rb))←Fn1(Ra, Rb)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
2-bit Dtype
4-bit Ra
4-bit Rb
1490
1491
1492
1493
1494
Rt(Fn0(Rb)) ←Fn2(Rb, 8Si) (8Si=8-bit Immediate Value)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
8-bit Immediate
2-bit Rb
1495
1496
1497
Rt(Fn0(Rb)) ←Fn3(Rb)
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
6-bit Opcode
6-bit Opcode Extension
4-bit Rb
1498

Fig. 14C

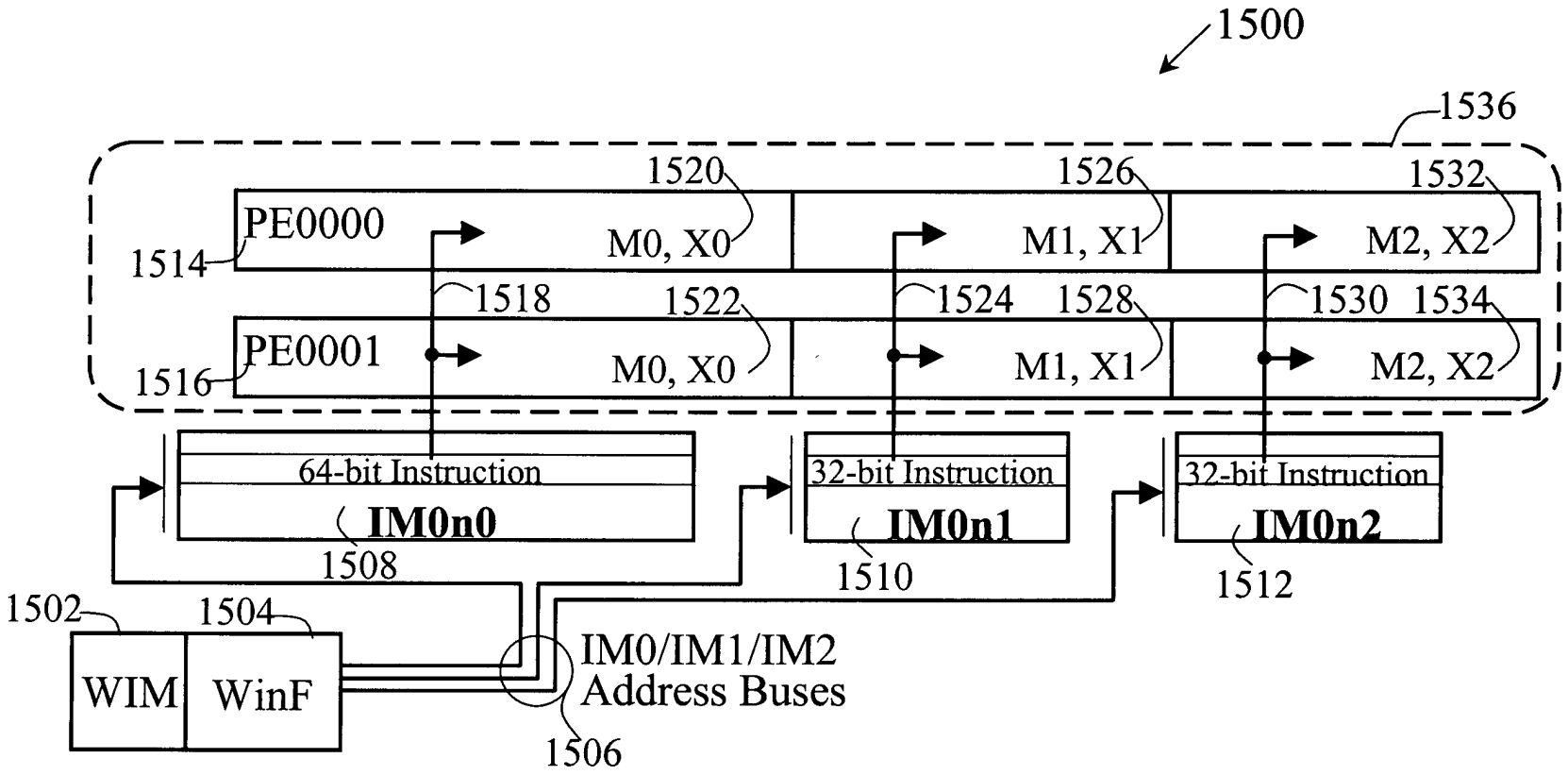

1500
1536
1520
1526
1532
PE0000
M0, X0
M1, X1
M2, X2
1514
1518
1522
1524
1528
1530
1534
PE0001
M0, X0
M1, X1
M2, X2
1516
64-bit Instruction
IM0n0
32-bit Instruction
IM0n1
32-bit Instruction
IM0n2
1508
1510
1512
1502
1504
WIM
WinF
IM0/IM1/IM2
Address Buses
1506

Fig. 15A

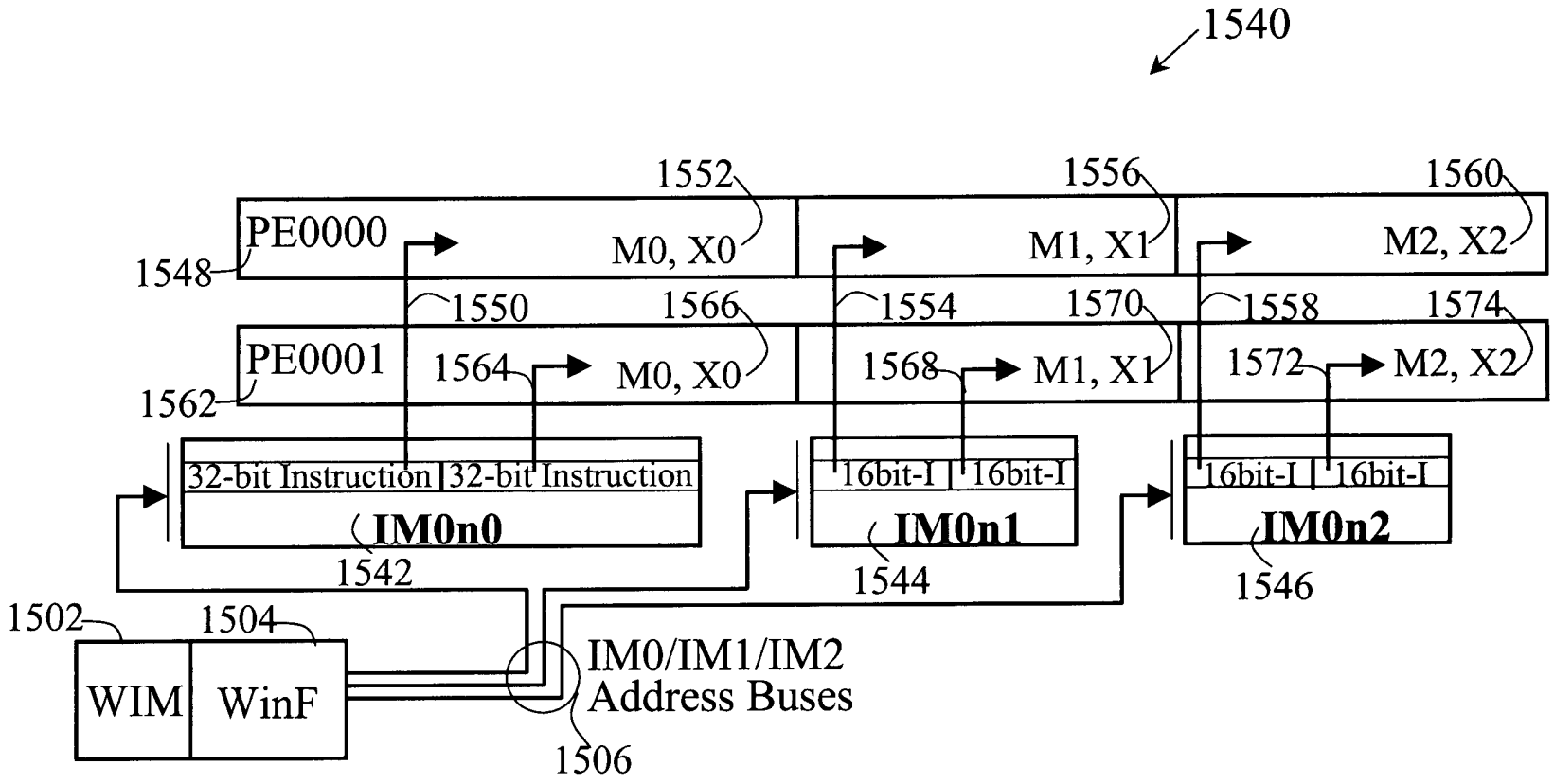

1540
1552
1556
1560
PE0000
M0, X0
M1, X1
M2, X2
1548
1550
1566
1554
1570
1558
1574
PE0001
1564
M0, X0
1568
M1, X1
1572
M2, X2
1562
32-bit Instruction
32-bit Instruction
IM0n0
16bit-I
16bit-I
IM0n1
16bit-I
16bit-I
IM0n2
1542
1544
1546
1502
1504
WIM
WinF
IM0/IM1/IM2
Address Buses
1506

Fig. 15B

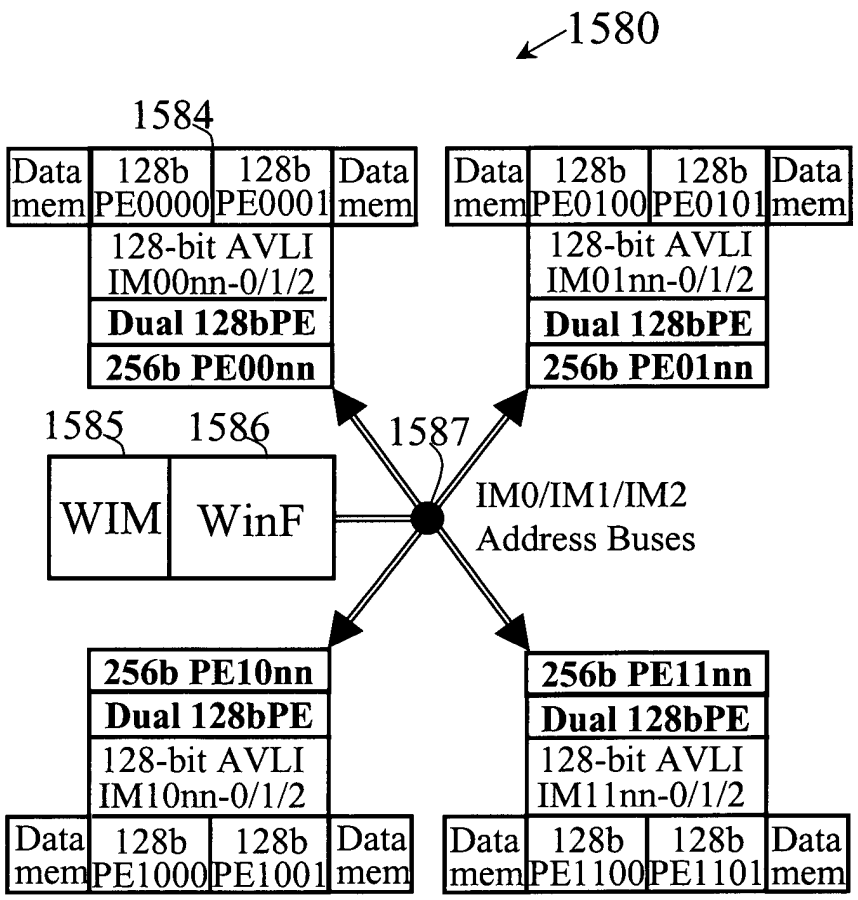

1580
1584
Data mem
128b PE0000
128b PE0001
Data mem
128-bit AVLI IM00nn-0/1/2
Dual 128bPE
256b PE00nn
Data mem
128b PE0100
128b PE0101
Data mem
128-bit AVLI IM01nn-0/1/2
Dual 128bPE
256b PE01nn
1585
1586
1587
WIM
WinF
IM0/IM1/IM2 Address Buses
256b PE10nn
Dual 128bPE
128-bit AVLI IM10nn-0/1/2
Data mem
128b PE1000
128b PE1001
Data mem
256b PE11nn
Dual 128bPE
128-bit AVLI IM11nn-0/1/2
Data mem
128b PE1100
128b PE1101
Data mem

Fig. 15C

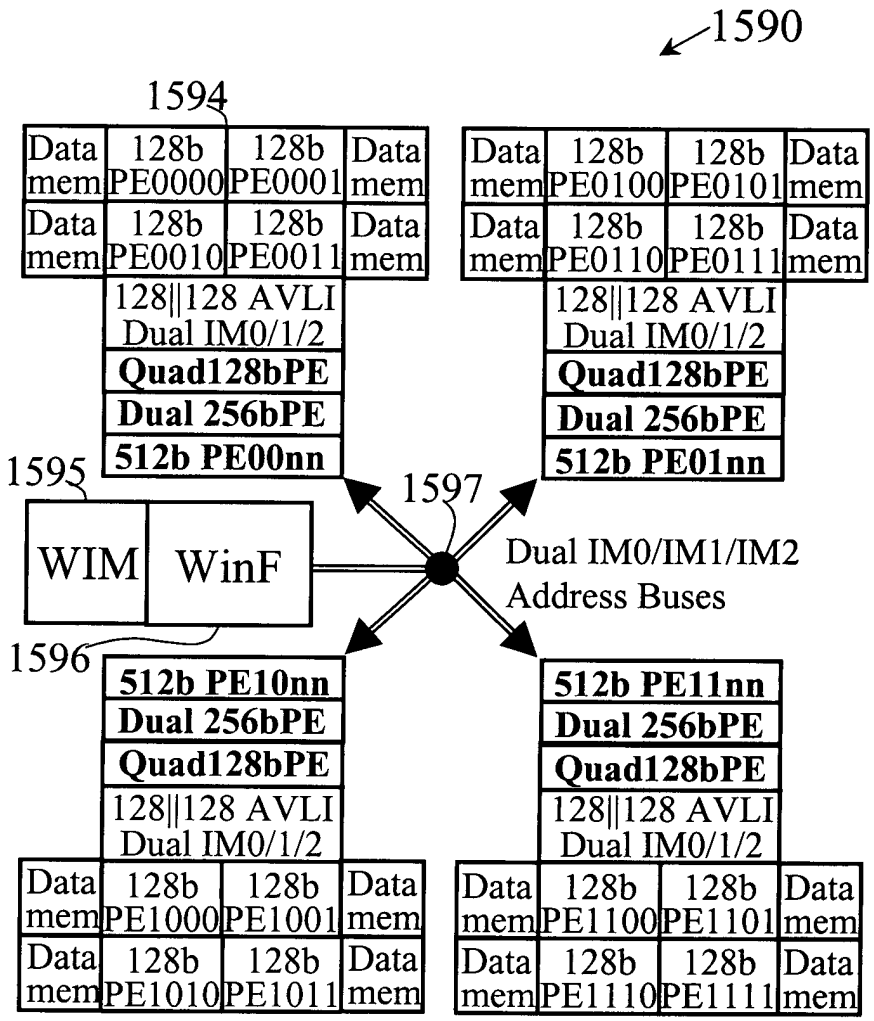

1590
1594
Data mem
128b PE0000
128b PE0001
Data mem
Data mem
128b PE0010
128b PE0011
Data mem
128||128 AVLI Dual IM0/1/2
Quad128bPE
Dual 256bPE
512b PE00nn
Data mem
128b PE0100
128b PE0101
Data mem
Data mem
128b PE0110
128b PE0111
Data mem
128||128 AVLI Dual IM0/1/2
Quad128bPE
Dual 256bPE
512b PE01nn
1595
WIM
WinF
1597
Dual IM0/IM1/IM2 Address Buses
1596
512b PE10nn
Dual 256bPE
Quad128bPE
128||128 AVLI Dual IM0/1/2
Data mem
128b PE1000
128b PE1001
Data mem
Data mem
128b PE1010
128b PE1011
Data mem
512b PE11nn
Dual 256bPE
Quad128bPE
128||128 AVLI Dual IM0/1/2
Data mem
128b PE1100
128b PE1101
Data mem
Data mem
128b PE1110
128b PE1111
Data mem

Fig. 15D